American Journal of Neural Networks and Applications
2022; 8(2): 12-16
http://www.sciencepublishinggroup.com/j/ajnna
doi: 10.11648/j.ajnna.20220802.11
ISSN: 2469-7400 (Print); ISSN: 2469-7419 (Online)

The Optimal Choice of Hybrid Convolutional Neural Network Components

Viktor Sineglazov, Illia Boryndo

Automation and Computer Integrated Complexes, National Aviation University, Kyiv, Ukraine

Email address:
svn@nau.edu.ua (V. Sineglazov), ibo.mistle@gmail.com (I. Boryndo)

To cite this article:
Viktor Sineglazov, Illia Boryndo. The Optimal Choice of Hybrid Convolutional Neural Network Components. *American Journal of Neural Networks and Applications*. Vol. 8, No. 2, 2022, pp. 12-16. doi: 10.11648/j.ajnna.20220802.11

Received: July 4, 2022; **Accepted**: July 25, 2022; **Published**: August 4, 2022

Abstract: Based on the development vector of modern AI systems and extensional complexity grows of analytical and recognition tasks it is concluded that the perspective class of convolutional neural networks – hybrid convolutional neural networks. Based on research results it’s proved that this type of neural networks permits to supply less mean square error under less overall structure complexity. The generic structure of hybrid convolutional neural network was proposed. It is shown and proved that these networks must include beside traditional components (convolutional layers, pooling layers, feed-forward layers) as well as an additional supportive layers (batch normalization layer, 1x1 convolutional layer, dropout layer, etc.) to achieve best both accuracy and performance results. The important properties of additional supportive layers (blocks) have been determined and researched. Based on the architectural requirements it is considered that modern topologies of hybrid CNNs are the combination of substantive CNNs such as Squeeze-and-Excitation neural network, poly-inception neural network, residual neural network, densely connected neural network, etc. It is listed the performance testing and final accuracy results for each block used both separately and in pairs to highlight it inner parameters, advantages and limitations. It is proposed an example of hybrid convolutional neural network constructed of investigated structural blocks. Calculated average training time shorten based on each of functional blocks, their advantages and integration details.

Keywords: Hybrid Neural Networks, Convolutional Neural Networks, Neural Network Performance, Dynamic Neural Network Structure, Image Representations

1. Introduction

Nowadays, in modern world fully filled with image processing tasks as well as the application of CNNs to solve them, the problems with lack of accuracy, low performance and network over-complexity issues occurs. The urgency of this problem over time is only increasing due to the proliferation of the problem of digital identification.

Most of CNN architectures are highly restricted due to their performance results, low learning rates and at the same time they require a big number of high quality training materials. In order to solve this problem and increase results the complexity of convolutional neural networks were only increasing over the years. But such development leads to new problems when further CNN enrichment bumps into hardware limitations. In such conditions hybrid CNNs takes the place. To increase overall performance and accuracy the group of CNNs can be combined to form hybrid convolutional neural network. In this paper we will describe and present the research results on the usage and topology features of hybrid convolutional neural networks (HCCN) [1] as well as different structural blocks that should be used for their synthesis processes. It includes performance research of each structural blocks, analysis of modern CNN topologies and their application using different learning samples to solve most of the performance issues.

The main criteria of this research are to define optimal types and structures of modern CNNs, extract functional blocks and apply them in the process of HCNN synthesis to achieve suitable performance and accuracy results.

2. Topology Analysis of Modern Convolutional Neural Networks

Nowadays, the active development of convolutional neural networks is still ongoing and new architectural solutions and ways to use them being actively put forward. Most of modern convolutional neural network architectures instead of simply focusing on increasing depth and structure complexity switched on using special structural blocks. These blocks [2] represent the unique combination pattern of simple layers, connectivity approach and usage of global or local assistance functions. These blocks by itself can provide superior improvements by applying it singly or as a combination of several different blocks.

Before diving into the blocks details, firstly it's necessary to take look on the modern CNNs and their advantages and disadvantages.

2.1. Channel Boosting Based Convolutional Neural Network

The training process performance of CNN also relies on the input representation and its parameters. The lack of different qualities and lack of class-specific information within target sample may affect CNN performance. To solve this problem, the concept of channel boosting CNN was designed. It based on the technology of input channel dimension and by using auxiliary learners that were introduced in CNN to boost the representation of the network. [4, 7]

Specifications of Channel Boosted CNN using Transfer Learning:

1) increasing representational capacity of CNN by means of rising quantity of input channels;
2) inductive transfer learning is used in a novel way to build a boosted input representation for CNN [7];
3) increases in computational load may happen due to the generation of auxiliary channels.

2.2. Feature-Map Exploitation Based Convolutional Neural Network

The main feature is as deep learning topology was extended, the more features maps were generated at each step. Many of the feature maps might be important for classification task, others might redundant or less important. [3]

Specifications of Squeeze and Excitation Network:

1) usage of block-based structural concept;
2) introduced SE structural block that can be integrated into many others CNN models due to its simplicity;
3) squeezes less important features and vice versa;

Specifications of Competitive Squeeze and Excitation Network:

1) uses feature-map wise statistics from both residual and identity mapping based features;
2) usage of both identity and residual feature-maps creates a reciprocal rivalry;
3) cannot be used in pair with attention-based concepts or structural blocks.

2.3. Attention Based Convolutional Neural Networks

Attention Networks implements the ability to choose which patch is the area of the focus or most important if the image. Let's take a view on some of them and their advantages:

Specifications of Residual Attention Neural Network (RANN):

1) generates attention aware feature-maps;
2) easy to scale up due to residual learning;
3) provides different representations of the focused patches;
4) adds soft weights on features using bottom up top-down feedforward attention;
5) superfluous complexity.

Specifications of Convolutional Block Attention Module (CBAM):

1) CBAM is a basic attention-based block designed for feed forward CNNs;
2) generate both feature-map and spatial attention in a sequential manner;
3) channel attention maps help what to focus;
4) spatial attention helps where to focus;
5) increases efficient flow of information;
6) simultaneous usage of both global avg pooling and max pooling.

2.4. Multipath Based Convolutional Neural Network

The main idea is to supply "shortcut" paths to skip some layers if it's necessary. Some of these shortcut connections are the following: zero padded, projection, dropout, 1x1 connections, etc. [6]

Specifications of Densely Connected Neural Network [13]:

1) introduced depth or cross-layer dimension;
2) ensures maximum data flow between the layers in the network;
3) avoid relearning of redundant feature-maps;
4) low and high level both features are accessible to decision layers;

3. Topology Analysis of Modern Convolutional Neural Networks

First of all, to understand the essence and major structural and functional advantages of hybrid CNNs it is necessary to look onto its structural unit – simple non-specific convolutional neural network. Functionally CNN could be represented as program-based logical unit that is used to process and analyze consumed graphical data by means of extracting the key features that contains the most valuable information in it. This evaluation is based on the validity of training samples and nested pre-trained behavior. Basically, CNN consists of two main parts: first one is extracting key features and second one is processing them and classifying due to initial task requirements. [5]

In the same way hybrid CNN essentially is the combination of two or more different CNNs that are configured and structured to work in pair (parallel or serial).

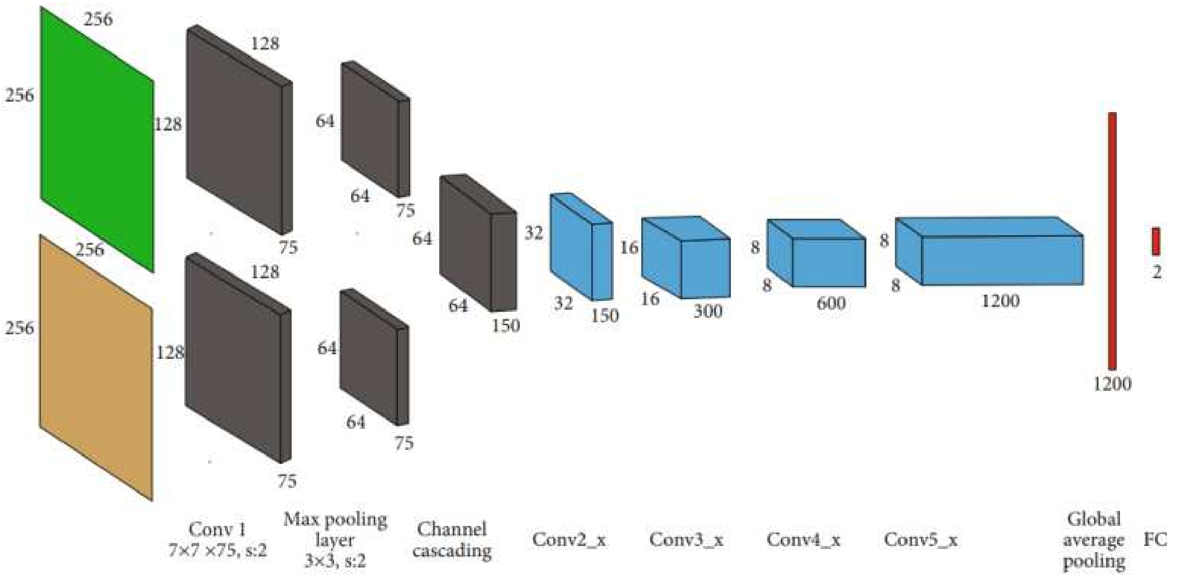


Figure 1. *HCNN simplified structural scheme.*

Functionally, HCNN architecture is an implementation of module-based pattern where each component of the model is responsible for performing only one specific task assigned to it. [8] These structural components could be not only inner CNNs but also supportive blocks, classifiers or specific task-related algorithms for graphical data preparation.

Therefore, applied task should be decomposed to the number of simple specific tasks which will be further processed by elements mentioned above. Such decomposition will increase overall network flexibility and will make it easier to train while boosting the positive effect on resulting accuracy and performance.

As an example of HCNN, the combination of densely connected and squeeze-and-excitation (SE) convolutional neural networks based onto ResNeXt structure could be presented. Here SE-CNN supplies model with global information accessibility through all structural layers as well as DenseNet prevents the information loses and increasing overall performance and accuracy. [9]

To merge different CNN architectures with their own features without losing their functional advantages the number of parameters should be considered:

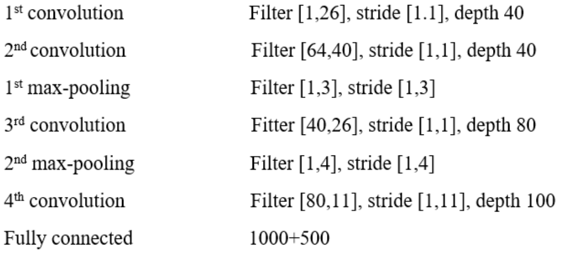
1) input data parameters such as initial scale, resolution, number of channels, sample size and recognition task type;
2) the output value of previous structural element (CNN, block, transitive layer) should be acceptable by the next one, initial target information should be saved;
3) the structures of both CNNs should be flexible and able to include supportive layers such as normalization layers, residual blocks, dropout layers, 1x1 convolution layers, etc.

By following mentioned rules, any type of CNN and structural blocks can be paired. It's based mostly on the specifics of tasks and input data parameters (e.g. image resolution, scale, color channels, number of training samples, etc.).

4. Functional CNN Modules and Optimization Based Construction Blocks

Based on listed number of different modern CNNs we can analyze them, extract their unique structural blocks and apply them to generate our own unique CNN architecture. Since these blocks has their own conceptual structure and features they should be analyzed as independent structural units as well as a different pair of such blocks placed together or separately within one neural network. Then it is necessary to apply performance testing to investigate their internal parameters, influence on overall system performance and accuracy shifts.

For our practical investigation we'll test these blocks by adding them to simply structured convolutional neural network. The simplicity will diminish external factors, randomness and allow us to clearly highlight the internal influence caused by each of the block. Described convolutional neural network model is presented on the Figure 2 and will be populated with obtained functional blocks. All the tests will be done using training and testing sample called "CIFAR-100". It contains low resolution images within the number of classes that fits well to process them using relatively simple system.



1st convolution	Filter [1,26], stride [1.1], depth 40
2nd convolution	Filter [64,40], stride [1,1], depth 40
1st max-pooling	Filter [1,3], stride [1,3]
3rd convolution	Fitter [40,26], stride [1,1], depth 80
2nd max-pooling	Filter [1,4], stride [1,4]
4th convolution	Filter [80,11], stride [1,11], depth 100
Fully connected	1000+500

Figure 2. *Initial test structure of performative CNN.*

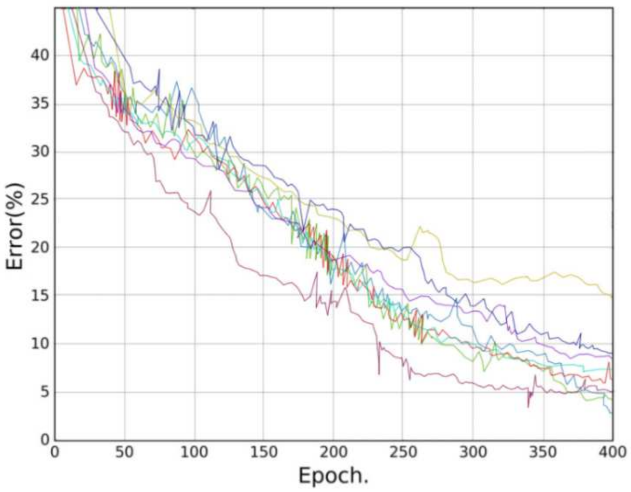


Figure 3. *Training cures of single-placed structural blocks (error rates).*

After applying given CNN to learn and process these images, the initial accuracy of such system is 86.3% and learning process lasts 5.3 hours. It will be the initial values for further performance test comparison. Take note that the overall learning time depends on the hardware and only the time differences should be taken into consideration. By adding structural blocks to the basic CNN system the result numbers were changed as following:

1) Densely connected layer (take note that this block may perform differently based on overall system depth) – accuracy 89,45%, learning process lasts 5.88h (performance boost = -10.94%);
2) SE-ResNet – accuracy 88,5%, learning process lasts 5.1h (performance boost = +3.8%);
3) SE-ResNeXt – accuracy 89,2%, learning process lasts 5.5h (performance boost = -3.77%%);
4) SE-BN-Inception – accuracy 87,92%, learning process lasts 4.92 h (performance boost = +7.17%);
5) Convolutional block attention module (take note that this block may perform better with the image samples of high resolution) – accuracy 92,1%, learning process lasts 5.42h (performance boost = -2.26%);
6) ResNeXt module – accuracy 87,81%, learning process lasts 5.51h (performance boost = -3.79%);
7) PolyInception module – accuracy 89%, learning process lasts 6.2h (performance boost = -16,98%);
8) Non-local Block – accuracy 89,45%, learning process lasts 5.88h (performance boost = -10.91%);

The combination of multiple blocks applied to following CNN structure gives us following results:

1) Dense + SE-ResNet – accuracy 94,25%, learning process lasts 6.9h (performance boost = -30.18%);
2) SE-ResNet + SE-ResNeXt – accuracy 91,8%, learning process lasts 6.6h (performance boost = -24.52%);
3) CBAM + Non-local block – accuracy 92,4%, learning process lasts 6.17h (performance boost = -16.41%);
4) CBAM + Dense – accuracy 94,28%, learning process lasts 6.31h (performance boost = -19.05%);
5) CBAM + PolyInception – accuracy 91.1%, learning process lasts 6.05h (performance boost = -14.15%);
6) Dense + ResNeXt – accuracy 90,88%, learning process lasts 5.97h (performance boost = -12.64%);
7) Dense + Non-local – accuracy 91,13%, learning process lasts 5.74h (performance boost = -8.3%);
8) Dense + PolyInception – accuracy 91%, learning process lasts 6h (performance boost = -13.2%);
9) Attention merge + CBAM – accuracy 94,4%, learning process lasts 6.34h (performance boost = -19.6%);
10) Attention merge + Dense – accuracy 92,5%, learning process lasts 5.5h (performance boost = -3.77%);
11) Attention merge + PolyInception – accuracy 89,9%, learning process lasts 5.65h (performance boost = -6.6%);

Occasionally, while combining several CNNs into one system the performance issues will occur causing low learning process results. There a number of solutions to deal with this, the most common is to populate CNN structure with supportive layers and blocks. The most common among them are:

1) batch normalization layer;
2) 1x1 convolution layer;
3) dropout layer;
4) residual block.

Batch normalization layer is the method of increasing network stability and performance. The main idea is that for several layers of CNN were given previously preprocessed data with zero mathematical expectation and zero dispersion.

By itself, the batch normalization is independent layer and could be applied at any stage of sequential model as a connector between its general layers (frequently placed right after pooling and convolutional layers). In the result it also can be used to avoid model overfitting problem. [11]

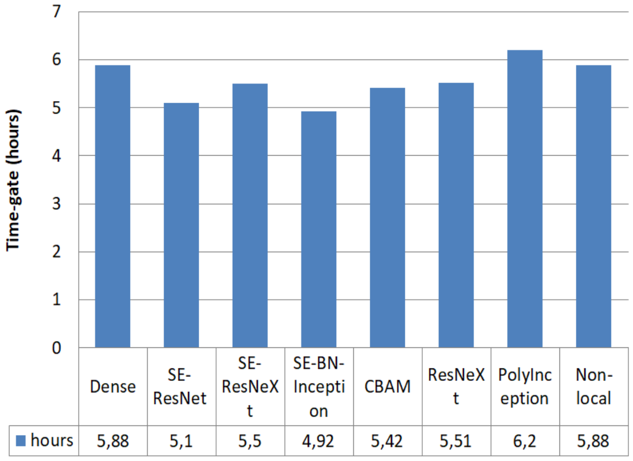


Figure 4. *Time-gate comparisons spent on complete learning process.*

1x1 convolutional layer essentially is a general convolutional layer with feature filter size equals to 1x1. One of its functions here is to reduce the dimension and reduce the amount of computation. The smaller the number of feature map channels, the smaller the amount of computation required for the convolution operation. But, as with image compression, the higher the compression ratio, the more frames are lost. So 1x1 convolution creates a linear combination of kernels to get in result more or less of equivalent kernels. [10] Another way to use it is for creation of a one-to-one projection of the feature maps to pool features across channels or to increase the number of feature maps, such as after traditional pooling layers.

Dropout layers is the tool used for preventing model overfitting. Functionally, dropout layers create random connections for number of neurons between the model layers. When the neurons are switched off the incoming and outgoing connection to those neurons is also switched off. In the result, it increases the learning process speed and prevents negative deviations that could reduce final accuracy. Dropouts are usually advised not to use after the convolution layers, they are mostly used after the dense layers of the network. It is always good to only switch off the neurons to 50%. If we switched off more than 50% then there can be chances when the model leaning would be poor and the predictions will not be good. [12, 13]

Finally, residual blocks are the blocks that presented with ResNet model and implements the “skip-connection” design pattern. Here, learning of residual functions is performed with reference to layer input instead of unreferenced learning. In the result it makes easier to optimize the referenced residual mapping compared to the original unreferenced mappings. Having skip connections allows the network to more easily learn identity-like mappings and increases overall model stability and learning speed.

5. Conclusions

In this paper it is defined the problems of structural synthesis of hybrid convolution neural networks (HCNN), it’s configuration and training process. Based on analysis of modern CNN architectures it is shown and described the necessity of different supportive and structural layers (blocks) that could enhance model parameters, increase overall performance, resulting accuracy and prevent model overfitting. At the end, the following elements were proposed: batch size: 512, 1x1 convolution layer, dropout layer, residual block. The number of specified functional blocks were tested in terms of performance and the results are listed at table 1. Based on this results it is concluded that “SE-ResNet” and “SE-BN-Inception” blocks are the best choice in terms of performance boost with relatively good resulting accuracy values. Using this blocks as an additive for any custom HCNN architecture will increase overall parameters and will enrich the model with peculiarities such as global cross-layer metadata accessibility, referenced residual mappings, etc. It is considered the importance of buffered transition of two HCNNs in overall model based on learning process comparison of raw model and enriched ones shown in Figure 3. The transitive elements will provide the backward compatibility for input/output of two nester neural networks.

Table 1. *Result parameter comparison table of single-used blocks.*

Block type	**Accuracy (%)**	**Time (H)**	**Diff (~)**
Densely connected layer	0.8945	5.88	3.2
SE-ResNet block	0.885	5.1	2.2
SE-ResNeXt block	0.892	5.5	2.6
SE-BN-Inception module	0.8792	4.92	1.8
Convolutional block attention module	0.921	5.42	5.9
ResNeXt module	0.8781	87.81	1.7
PolyInception module	0.89	89	2.7
Non-local Block	0.8945	89.45	3.1

References

[1] Iveta Mrazova, Marek Kukacka. "Hybrid convolutional neural networks" IEEE International Conference on Industrial Informatics 10.1109/INDIN.2008.4618146.

[2] Chaitanya Nagpal and Shiv Ram Dubey. “A Performance Evaluation of Convolutional Neural Networks for Face Anti Spoofing”.

[3] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014, pp. 675–678.

[4] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014.

[5] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

[6] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu. “Squeeze-and-Excitation Networks”.

[7] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, D. Ramanan, and T. S. Huang, “Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks,” in ICCV, 2015.

[8] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in ICCV, 2015.

[9] L. Shen, Z. Lin, and Q. Huang, “Relay backpropagation for effective learning of deep convolutional neural networks,” in ECCV, 2016.

[10] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten, “Exploring the limits of weakly supervised pretraining,” in ECCV, 2018.

[11] G. Mittal, S. Sasi, “Robust Preprocessing Algorithm for Face Recognition”, Proceedings of the 3rd Canadian conference on Computer and Robot vision, United States of America, 2006.

[12] Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems”, pages 1097–1105, 2012.

[13] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger, “Densely Connected Convolutional Networks”, 2018, arXiv: 1608.06993.